

Refresh Tokens

Description

In this exercise you'll learn how to obtain a refresh token and use it to get new access tokens.

Estimated Duration

15

Instructions

Make sure you've completed the first Getting Started exercise, as you'll need the account and setup steps in that exercise to be complete first.


The goal of this exercise is to get a refresh token and use the refresh token to get a new access token. We will be building on the previous exercise where you used the authorization code flow to get an access token. Rather than repeat all the setup steps here, we'll assume you have already done that exercise.

From the side menu, click **Applications** and then **Applications**.

The screenshot shows the Okta management interface. On the left, the sidebar menu has 'Applications' selected. The main area displays a notification: 'Your plan provides a limited number of custom apps.' with a link to the 'plan page' and an 'Upgrade' button. Below the notification are four buttons: 'Create App Integration', 'Browse App Catalog', 'Assign Users to App', and 'More'. A table lists installed applications with columns for 'STATUS' and a count. The table shows four applications: 'My Web App' (ACTIVE, 1), 'Okta Admin Console' (INACTIVE, 4), 'Okta Browser Plugin' (INACTIVE, 4), and 'Okta Dashboard' (INACTIVE, 4). A search bar is located above the table. At the bottom left, a message says 'Thanks for trying the Okta Starter plan. Upgrade to the Advanced plan to create more apps and get more Monthly Active Users.' with an 'Upgrade' button.


STATUS	
ACTIVE	1
INACTIVE	4

Then select your application you created previously.



My Web App

Active

 [View Logs](#)

General

Sign On

Assignments


Okta API Scopes

Client Credentials

Edit

Client ID


Ooaa10fbKOxfzjGb5d6



Public identifier for the client that is required for all OAuth flows.

Client secret

.....




Secret used by the client to exchange an authorization code for a token. This must be kept confidential! Do not include it in apps which cannot keep it secret, such as those running on a client.

General Settings

Edit

Okta domain

dev-22588100.okta.com



APPLICATION

Application name

My Web App

Application type

Web

Allowed grant types

Client acting on behalf of itself

☐ Client Credentials

Client acting on behalf of a user

☒ Authorization Code

Ready to code

You can download a preconfigured sample app.

[Download sample app](#)

To get started using your custom app integration, see the "Sign Users In" section in the Okta [Developer's guide](#)

Click on **Edit** under **General Settings**, then scroll down and enable the **Refresh Token** checkbox.

APPLICATION

App integration name

My Web App

Application type

Web

Grant type

Client acting on behalf of itself

☐ Client Credentials

Client acting on behalf of a user

☒ Authorization Code

☒ Refresh Token

☐ Implicit (Hybrid)

This allows your application to request refresh tokens and use them. Without this checked, the authorization server will not issue refresh tokens to this application.

Now you'll want to start a new OAuth flow and request a refresh token. Build the authorization URL like you did in the previous lesson, but this time also add the scope `offline_access` to the request.

```
https://dev-xxxxxx.okta.com/oauth2/default/v1/authorize?
  response_type=code&
  scope=offline_access+{YOUR_SCOPE}&
  client_id={YOUR_CLIENT_ID}&
  state={RANDOM_STRING}&
  redirect_uri=https://example-app.com/redirect&
  code_challenge={YOUR_CODE_CHALLENGE}&
  code_challenge_method=S256
```

Paste the completed URL into the Refresh Token exercise

(<https://oauth.school/exercise/refresh/>) to check your work. This will double check that you've included the right scope in the request. Once that's confirmed, the "Log In" button will appear. Click that and you'll be taken to the authorization server, and since you're already logged in, you'll be redirected back immediately with an authorization code in the query string.

example-app.com/redirect?code=_tjr07noWymenvuquqwoLQb9oQPKnfEeAlfKSy26u6o&state=34134

Congrats!

The authorization server redirected you back to the app and issued an authorization code!

You can exchange this authorization code for an access token now!

Your app can read the authorization code and state from the URL, and they are printed below for your convenience as well.

```
code=_tjr07noWymenvuquqwoLQb9oQPKnfEeAlfKSy26u6o
state=34134
```

You should verify that the state parameter here matches the one you set at the beginning. Otherwise it's possible someone is trying to trick your app!

Now you'll need to make a POST request to the token endpoint to get an access token. This request is the same as before. Replace the placeholder values with your own.

```
curl -X POST https://dev-xxxxxx.okta.com/oauth2/default/v1/token \
  -d grant_type=authorization_code \
  -d redirect_uri=https://example-app.com/redirect \
  -d client_id={YOUR_CLIENT_ID} \
  -d client_secret={YOUR_CLIENT_SECRET} \
  -d code_verifier={YOUR_CODE_VERIFIER} \
  -d code={YOUR_AUTHORIZATION_CODE}
```

If everything worked, you'll get back a response that includes both an access token as well as a refresh token! Paste the entire token response (not just the access token) into the [oauth.school](#) website to check your work.

Great! You got a refresh token! Now use it to get a new access token, and paste the new response from the token endpoint below.

Refresh Token Response

```
{  
  "token_type": "Bearer",  
  ...  
}
```

Use the refresh token to get a new access token, then paste the entire token response JSON here to check your work

Check Your Response

Reset

If that succeeds, you'll be taken to the next step. Now you'll need to use the refresh token to get a new access token.

Make a POST request to the token endpoint again, but this time you'll use new parameters to make the refresh token request.

```
curl -X POST https://dev-xxxxxxx.okta.com/oauth2/default/v1/token \  
-d grant_type=refresh_token \  
-d client_id={YOUR_CLIENT_ID} \  
-d client_secret={YOUR_CLIENT_SECRET} \  
-d refresh_token={YOUR_REFRESH_TOKEN}
```

You should get back a new access token response, which will look similar to the previous response except this will include a new access token. Paste the entire response into the field to check the result!

If that worked, you'll get a message saying you've completed the exercise!